

Desarrollo y difusión de un método para el diseño de la base de datos a partir del modelo orientado a objetos.

Autores: M.Sc. Anaisa Hernández González (*)

Dra. Sofía Álvarez Cárdenas ()**

Instituto Superior Politécnico José Antonio Echeverría (ISPJAE)

Centro de Estudios de Ingeniería de Sistemas

Teléfono: 537-2607912 y 537-2606418

Fax: 537-2671575 y 537-2671574

(*) anaisa@ceis.ispjae.edu.cu

() sofia@ceis.ispjae.edu.cu**

Resumen.

La mayoría de las aplicaciones de software que se desarrollan en el mundo requieren del almacenamiento y gestión de grandes volúmenes de información. Con el auge del paradigma de la orientación a objetos, este proceso ha tomado nuevas dimensiones porque la persistencia es de objetos, no solo de datos. En este trabajo se describe un método para el diseño de la base de datos a partir del paradigma de la orientación a objetos (**DIBAO**), desarrollado en nuestro centro.

Se incluyen las características principales de una herramienta CASE, que automatiza el diseño y la generación de la estructura estática hacia SQL Server, Paradox y Oracle 8.0, y los beneficios que aporta su uso en el desarrollo de un producto de software, por la disminución en el tiempo de desarrollo que implica.

Al finalizar se comentan las experiencias que se tienen con el uso del método y la herramienta automatizada.

Palabras claves: Base de datos, Ingeniería de Software.

Introducción

Diseñar la base de datos (BD) a partir de un modelo orientado a objetos (MOO) no debe hacerse aplicando los métodos tradicionales porque no se trabaja con los mismos conceptos, por lo tanto, se requieren de nuevos métodos y herramientas de desarrollo.

Gran parte de las metodologías desarrolladas que siguen el enfoque orientado a objetos (OO), eluden el tema del diseño de la base de datos. Incluso hay algunas que reconocen que no abordan este problema. Esta situación era lógica si se tiene en cuenta el estado de los gestores de objeto, y que el desarrollo de métodos para un nuevo enfoque va precedido del desarrollo de lenguajes que lo soporten. La tendencia más explotada ha sido convertir las clases a tablas o almacenar la información utilizando las potencialidades de los lenguajes de programación orientados a objetos (LPOO) para guardar los valores de los atributos de los objetos.

En éste trabajo describimos algunas herramientas y procedimientos para obtener el comportamiento estático y dinámico de los objetos y cómo almacenarlos en un medio persistente, como parte de un método para el diseño de la base de datos, al que se ha denominado **DIBAO**.

Tomando como referencia los aportes de acompañar los métodos en la Ingeniería de Software por herramientas automatizadas (CASE), que respalden su utilización y reduzcan el tiempo de desarrollo, se describe los esfuerzos en este caso realizados, materializados en el CASE Gebase.

Pero, todos estos esfuerzos de la universidad en el desarrollo de investigaciones en función de las necesidades de la sociedad, no serían suficientes si no existen y se implementan los mecanismos para transmitir y utilizar los resultados. Es por ello que este trabajo concluye con un bosquejo de lo que se ha hecho en este campo.

Método DIBAO.

El estado del arte en el campo de las BD, condiciona las alternativas del. El método para el Diseño de la BAs e de datos a partir del modelo orientado a Objetos (**DIBAO**), está compuesto por dos partes: un modelo de persistencia y una capa persistente de clases (Figura 1). El modelo de persistencia describe los pasos para el diseño de la BD, las herramientas que se deben utilizar en este proceso, cómo convertir las clases al medio de almacenamiento seleccionado y cómo interpretar la información contenida en los diagramas y especificaciones textuales, en función de este diseño. La capa persistente de clases incluye las subcapas de especificación (define las clases para describir las nuevas especificaciones asociadas a las relaciones entre los objetos) y de interfaz (define las clases que se encargan de la relación con el medio de almacenamiento).

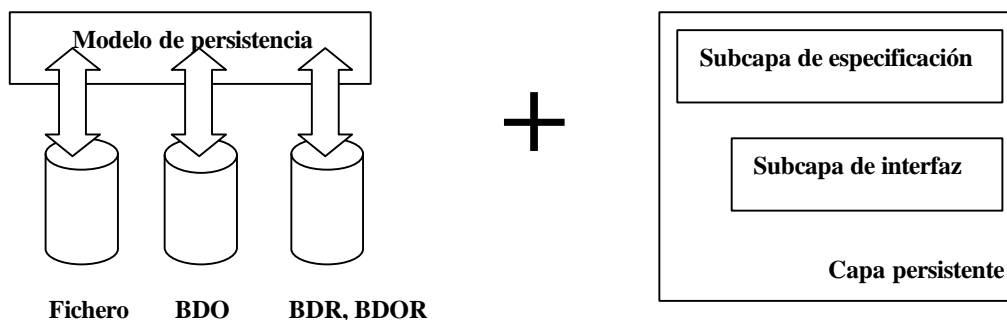


Figura 1 Estructura del método **DIBAO**.

Modelo de persistencia.

Independientemente del medio de persistencia seleccionado para almacenar los objetos persistentes, se deben realizar un grupo de pasos que permiten completar la semántica de los objetos en aspectos que son importantes referentes a su estructura estática y comportamiento dinámico. Los pasos que se proponen son:

1. *Definir las clases persistentes.*

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Es responsabilidad del diseñador definir cuáles clases son las que deben ser persistentes, aunque el método propone un conjunto de criterios que ayudan en este proceso, relacionados con los árboles de jerarquía y composición. [HER00a]

2. *Refinar las clases.*

El objetivo de este paso es la obtención de la jerarquía de clases y la definición de nuevas clases. No es obligatorio que aparezcan nuevas clases, se recomienda revisar si existe algún comportamiento de interés que no haya sido tomado en cuenta y sea trascendental en la solución del problema, para incluirlo.

3. *Clasificar las clases y los atributos.*

Clasificar las clases (simples, complejas y compuestas) en dependencia de los atributos que la integran y los atributos (estáticos, dinámicos y derivados) teniendo en cuenta si son o no afectados por los eventos.

4. *Realizar el diagrama de clases.*

La mayoría de las características de la perspectiva estática son capturadas gráficamente en el diagrama de clases (DC) cuya notación toma como base la del diagrama de estructura de UML, aunque se añaden nuevas características usando estereotipos y notas.

El DC incluye la definición de restricciones de integridad estática a través de las cardinalidades de las relaciones, la dimensión E/D y el tipo de datos asociado a cada atributo. Para el resto de estas restricciones, el método **DIBAO** sugiere que se especifiquen textualmente completando para cada atributo de cada clase: nombre, si toma valor único dentro de los objetos de esa clase, si puede tomar valor nulo, rango, valor por defecto y fórmula de derivación. Además, para todos aquellos atributos que tengan restricciones de dominio más complejas, hay que especificar las reglas que indiquen qué valores pueden tomar.

Se especifican la dimensión Estática o Dinámica, para las relaciones entre la clase compuesta y cada una de sus clases componentes, que se representa gráficamente usando estereotipos. Estas dimensiones, al igual que la clasificación de los atributos, se toma de la metodología OO-Method. [PAS96]

En las asociaciones entre clases se especifica en ambos extremos de la relación las cardinalidades máximas y mínimas (<máximas, mínimas>).

Al representar la herencia se debe indicar el atributo del padre y el valor que toma, cuando el padre se especializa en cada hija. Si éste atributo no existe en la definición de la clase padre, se añade. Además se especifica si las herencias que se derivan del padre son total/parcial y con o sin solapamiento. Total implica que todos los objetos del tipo de la clase padre se especializan en al menos un objeto de alguna clase padre, y solapamiento que al menos un objeto del tipo de la clase padre se especializa en dos o más objetos de clases hijas diferentes.

5. Realizar el diagrama de transición de estado.

Para mostrar la dinámica del comportamiento de un sistema, se ha hecho uso durante años de los diagramas de transición de estado (DTE). Este modelo recomienda utilizar los DTE en función del diseño de la BD, construyéndolos para aquellas clases que posean atributos dinámicos pues en función de estos es que están los posibles estados por los que transita un objeto, por lo que brinda recomendaciones para obtenerlo.

En el modelo se definen los principales elementos que componen un DTE y se brindan recomendaciones sobre:

?? Formato de transiciones.

?? Definición de estados agregados sobre la base de los eventos que afectan a un mismo atributo y el acoplamiento entre estados.

?? Reglas para el balance entre los niveles de diagramas, en cuanto a las transiciones de entrada y salida.

Se adopta la clasificación de los atributos dinámicos en cardinales, característicos de un estado y pertenecientes a una situación, propuesta en [PAS96]. En la metodología OO-Method se utiliza para obtener las fórmulas de evaluación de estos atributos, en este modelo se tomó como punto de partida en las recomendaciones que se brinda para identificar estados, como parte del proceso de construcción del DTE.

Del DTE se pueden obtener, para cada clase, las fórmulas dinámicas: precondiciones, disparadores y fórmulas de evaluación de los atributos que cambian su valor en el tiempo cuando ocurren determinados eventos (atributos dinámicos).

6. Obtener las restricciones estáticas y las fórmulas dinámicas.

Especificando textualmente o derivando a partir de los diagramas de clases y transición de estado.

7. Convertir las clases al medio de almacenamiento.

Es en este paso en el que se tiene en cuenta hacia qué lugar se guardarán los objetos. El factor decisivo en la selección depende de las aplicaciones y los datos que maneja, pudiendo utilizarse alguna de las siguientes variantes [HER97]:

?? Si se tiene un Sistema de Gestión de Base de Datos de Objetos, hay que garantizar que las definiciones obtenidas se correspondan con el modelo de objetos de ODMG.

?? Si lo que se tiene es un lenguaje de programación orientado a objetos, que permite relacionarse con una base de datos relacional (BDR) o un Sistema de Gestión de Base de Datos que mezcle conceptos relacionales y de la orientación a objetos, la solución es llevar las clases definidas a tablas.

?? Si lo que se tiene es un LPOO, que no permite relacionarse con una base de datos relacional, pero que permite salvar registros, entonces se deben utilizar las clases que brinda para almacenar los atributos de las clases persistentes en ficheros estructurados. Puede que el lenguaje permita la relación con una base de datos relacional, pero se escoge como forma de almacenamiento la organización en ficheros.

Capa persistente.

Cuando se trabaja con un gestor de objetos, internamente se implementan los conceptos del enfoque OO y las capacidades de las BD, por lo que conceptos como los relativos a la herencia, por ejemplo, son tratados automáticamente sin que intervengan los programadores. En las variantes de utilizar como medio de almacenamiento los ficheros o un LPOO que permita relacionarse con gestores relacionales u objeto/relacionales, será responsabilidad de diseñadores y programadores definir las clases que permitan manejar los conceptos de objeto y otros relacionados con las relaciones entre clases, que no están incluidos dentro de los modelos relacional y objeto/relacional.

La posibilidad de especificar clases permite la definición de una capa de clases persistente que encapsule el trabajo con los datos almacenados.

Cuando se trabaja con un LPOO, pero la información se almacena en un BDR o BD objeto/relacional, es posible encapsular el trabajo con el medio de almacenamiento en una estructura intermedia que sea la única afectada si cambia el gestor o la estructura de las tablas en que se almacena la información de un objeto [HER01]. Evidentemente esta solución hace más compleja la construcción de aplicaciones, pero crea una disciplina que mejora la calidad del producto final, facilita el mantenimiento y permite el desarrollo de aplicaciones más complejas.

Esta capa de clases incluye métodos para crear, modificar, buscar y eliminar un objeto. De manera que desde afuera (en su interfaz) no se conoce cómo realmente se almacenan los objetos (por ejemplo, cuántas tuplas de cuántas tablas) porque queda totalmente encapsulado. [HER01]

GeBase.

A partir de la definición del método de diseño se abordó la tarea de desarrollar una herramienta CASE (Computer-Aided Software Engineering) que generara automáticamente, la estructura estática de las bases de datos y los métodos que manipularán dicha base de datos, teniendo como principales objetivos:

1. Definir las propiedades que caracterizan estáticamente a las clases.
2. Generar la estructura estática de la Bases de Datos hacía SQL Server, Oracle y Paradox.
3. Generar Código Fuente (definición de clases y métodos en Object Pascal) para Insertar, Buscar, Eliminar y Modificar la Base de Datos generada hacía SQL Server.

En la herramienta obtenida se define la estructura estática completa y la capa de clases intermedia, entre las aplicaciones y el medio de almacenamiento persistente, que permita acceder a los datos almacenados. Esta capa se implementa en Object Pascal.

Esta herramienta fue implementada con el ambiente visual Borland Delphi 5.0 sobre Microsoft Windows 95/98 y Windows NT/2000/XP. A la herramienta se le dio el nombre de **GeBase** [PER01].

La aplicación está destinada, fundamentalmente, a los analistas de sistemas y aquellas personas que se adentren en el fascinante mundo de la programación orientada a objetos y que utilicen un Sistemas de Gestión de Bases de Datos Relacionales para el almacenamiento de los objetos.

Gebase genera la estructura estática de las bases de datos y el código fuente para la manipulación de dicha base de datos, o bien a partir de los resultados de la etapa de análisis, o a través de la creación de un proyecto nuevo, haciendo posteriormente un refinamiento de las estructuras ya almacenadas.

Otro aspecto importante que permite definir **Gebase** es la jerarquía de clases. De la definición de esta jerarquía surgirán nuevas clases y algunas se eliminarán, además de modificarse las propiedades de las existentes, específicamente sus atributos, cambiarán por la posición que poseen en la jerarquía.

La comunicación de **Gebase** con los ficheros del análisis, el flujo de información a través de ella, así como los servicios que presta y su propósito fundamental queda ilustrado en la Figura 2 donde se representa el contexto en el que se mueve la aplicación, y se puede apreciar que las salidas finales serán un fichero con estructura propia del sistema y extensión Gbd, conteniendo la estructura estática de las bases de datos, garantizando de esta forma que el usuario pueda recuperar dicho avance en cualquier momento que sea requerido y hacer uso de la información especificada, así como del código y las tablas que se generan.

La generación comprende diferentes procesos entre los que están:

- ?? Generación física de las tablas que formarán parte de la Base de Datos junto a sus llaves y campos
- ?? Generación de los métodos que realizarán las operaciones estándar para la manipulación de tablas de la Base de Datos. Junto a estos métodos también se generará el código de aquellos métodos proveniente del análisis que no manipulan la Base de Datos y cuyos datos serán especificados en la etapa de completamiento dentro del **Gebase**.
- ?? Generación de la documentación del Proyecto que se está desarrollando para Word.

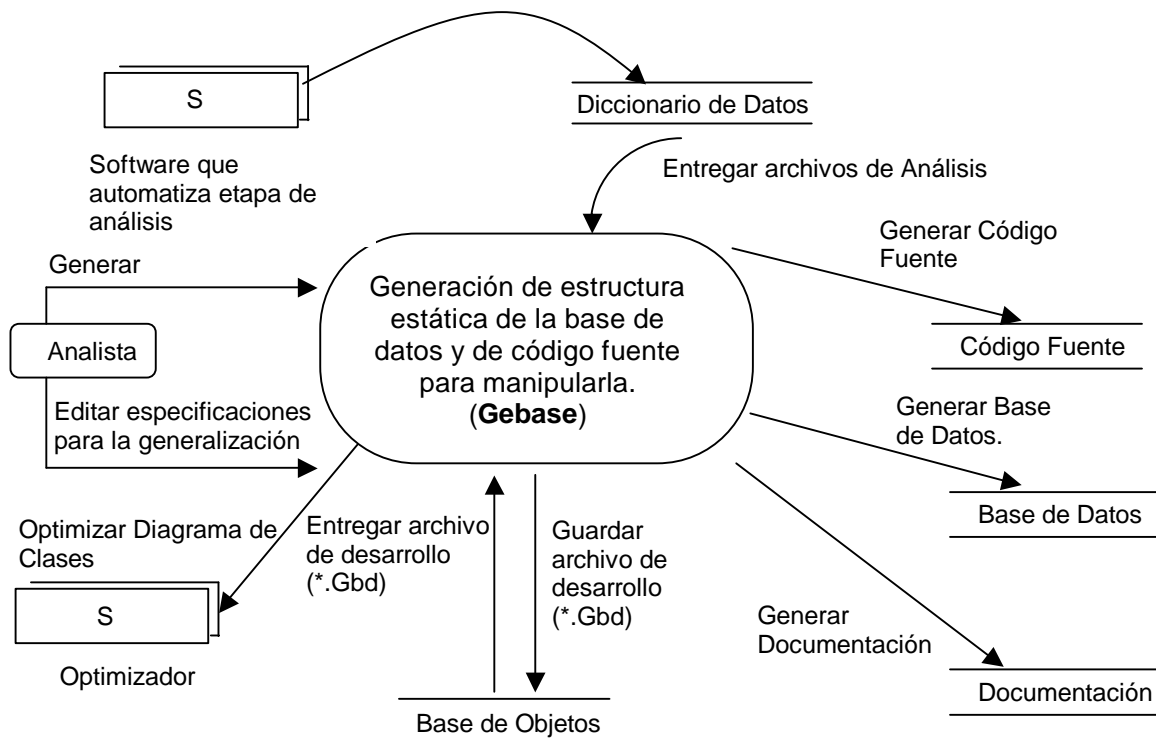


Figura 2 Relación de **GeBase** con el medio.

El sistema propuesto garantiza además:

- ?? La generación del diagrama de clases a partir de las características de estas y las relaciones entre ellas (asociaciones).
- ?? La adición en el diagrama de la semántica referente a la relación entre clases de herencia, agregación/composición y asociación.
- ?? Lograr una mejor ubicación espacial de los elementos que componen el diagrama mediante la utilización de un algoritmo de optimización basado en el Escalador de Colinas Estocástico.

Para lograr una interfaz de calidad que garantice un nivel de comunicación elevado se debe velar por aspectos como: la utilización de colores adecuados, frases de interpretación única, la tipografía, la disposición espacial de los elementos que la componen, entre otros. El sistema propuesto, Figura 3, está dotado de un ambiente similar al que proporciona Windows disminuyendo así el tiempo de aprendizaje empleado por los usuarios al interactuar con la aplicación.

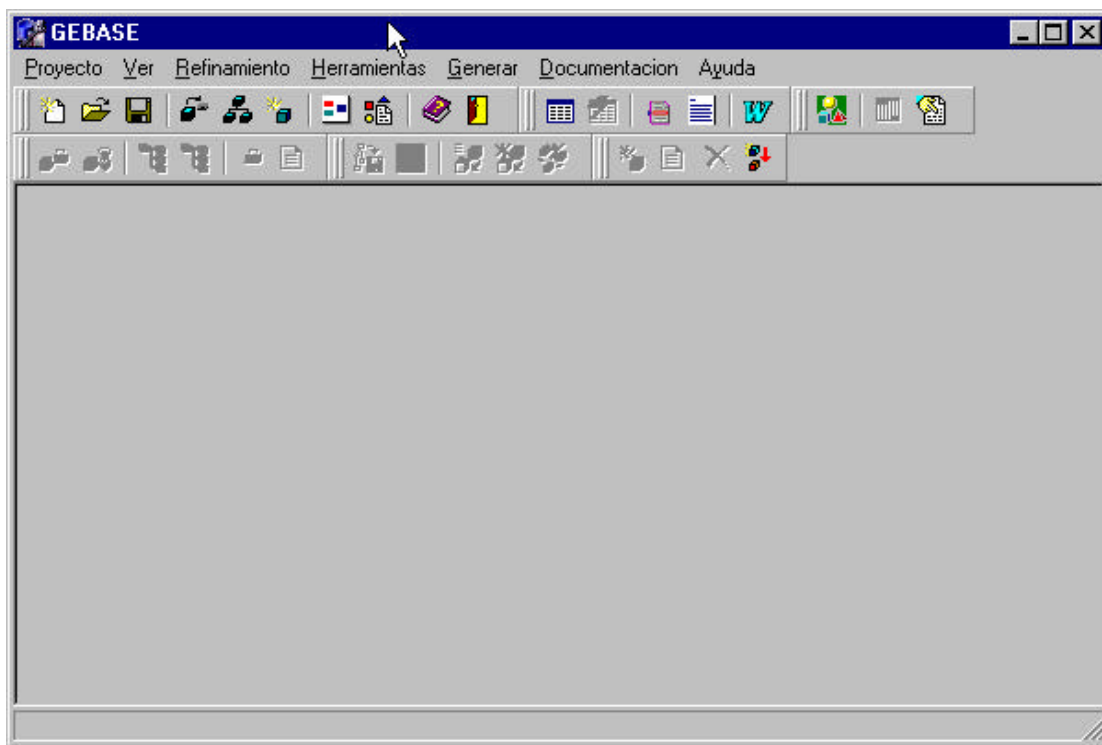


Figura 3 Ventana principal de la Aplicación

Las opciones del sistema son las siguientes:

- ?? Proyecto: se incluyen las funciones relacionadas con el manejo de un proyecto como crear un nuevo proyecto, abrir uno existente, abrir el explorador de proyectos, editar los tipos de datos del proyecto, exportar el diagrama de clases del proyecto y salir de la aplicación entre otros.
- ?? Ver: permite escoger entre cuales barras de herramientas ver y ocultarlas, así como si se desea trabajar con todas las clases del proyecto abierto en curso o solo las persistentes.
- ?? Refinamiento: se refinan las jerarquías de clases, las relaciones entre las clases, la persistencia y se adicionan y eliminan las clases.

- ?? Herramientas: incluye la eliminación e inserción de objetos tanto dentro del Explorador de clases como a la hora de la edición del Diagrama de Clases. Permite además optimizar el Diagrama de Clases, la inserción de clases y asociaciones, la eliminación de un objeto seleccionado y el reporte de un objeto si este es de tipo clase.
- ?? Generar: se genera la estructura de la Base de Datos las units con las clases para trabajar con las tablas generadas para SQL Server y la documentación para Microsoft Word.
- ?? Documentación: permite una rápida documentación de las clases, las tablas y el código.
- ?? Ayuda: incluye la ayuda estándar de Windows acerca del sistema y el "Acerca de" del **Gebase**.

Al automatizar algunos pasos de una metodología, **Gebase** ayuda a los desarrolladores pues los guía en el desarrollo de un proyecto y lo previene contra errores o inconsistencias en el diseño de la BD que afectan la calidad del proyecto final.

La documentación que genera para Microsoft Windows 95/98 y Windows NT/2000/XP, permite reducir en un 75% el tiempo dedicado a esta tarea. Además el formato en que se obtiene la documentación generada, facilita su inclusión como parte de la documentación técnica de la aplicación.

Gebase se estima que ahorra entre un 20%-25% del tiempo de desarrollo de un proyecto informático, de acuerdo al volumen de clases persistentes (mientras mayor es la cantidad con respecto al total de clases, se obtiene un ahorro mayor). Constituye además una ayuda para el diseñador pues ya no tendrá que cambiar el código cada vez que desee cambiar la estructura de la BD, solo tiene que cambiar la especificación y volver a generar.

Presentación del método utilizando libros.

Los libros permiten la comunicación de ideas y conocimientos entre autor y lector, no importa si su presentación es en papel impreso o como publicación electrónica. En el proceso de instrucción, que se lleva a cabo en cualquier nivel de enseñanza, las limitaciones que pueden existir en la impresión a gran escala y la difusión del libro en papel, en relación con los libros electrónicos; no pueden frenar la elaboración de estos importantes documentos que, cuando se conciben con fines docentes, son un arma imprescindible tanto para el profesor como para el alumno.

El método DIBAO forma parte de un libro [HER00b] elaborado por profesores de Informática del Instituto Superior Politécnico José Antonio Echeverría (Ms. C. Anaisa Hernández González y Lic. Rosa Ma Mato García) y el Instituto Técnico Militar José Martí (Ing. Daniel Tenorio), como texto docente de apoyo a las asignaturas relacionadas con los Sistemas de Gestión de Base de Datos y Análisis y Diseño de Sistemas Informáticos.

El libro “*Base de datos: coexistencia de los modelos relacional y orientado a objetos*”, pretende poner a disposición de sus lectores, un documento didáctico elaborado para enseñar y aprender. El texto recoge las experiencias acumuladas durante años en la enseñanza del modelo relacional y los aspectos más interesantes e importantes del MOO, resultado de la investigación de los autores y tema de tesis de doctorado de la autora principal del libro.

En cuanto al método **DIBAO**, el documento contiene las características principales, la definición de los conceptos asociados, la fundamentación teórica y cómo propone que se diseñe la BD.

El libro se hizo, en principio, destinado a los estudiantes y profesores de carreras informáticas, aunque por el contenido que incluye puede ser útil para todos aquellos que requieran que la

información con que trabajan sus aplicaciones trasciendan en el espacio y en el tiempo. Constituye un material de guía para los profesores de cursos de bases de datos y análisis y diseño de sistemas informáticos, independientemente del gestor que utilicen para ejemplificar sus explicaciones.

El libro se divide en tres partes, excepto la primera, el resto se subdivide en capítulos. Se incluyen además 2 anexos. La Parte III es la dedicada al MOO y en ella se describe en qué consiste este modelo y el estado de arte en la actualidad. Se explica cómo se implementan las capacidades de las bases de datos poniendo ejemplos de gestores presentes en el mercado. Al finalizar se enseña a diseñar la BD tomando como base un análisis OO y una implementación en diferentes plataformas. Se encuentra subdividido en los capítulos:

4. Introducción al modelo orientado a objetos.
5. Capacidades de las bases de datos orientadas a objeto.
6. Estado del arte en el diseño de la base de datos.
7. Soluciones al diseño de la base de datos

Al inicio de cada capítulo se proponen preguntas cuya respuesta se conoce a lo largo de sus páginas y que en las conclusiones se resume. Las preguntas no tienen una respuesta directa en el texto, para responderla es necesario que los estudiantes asimilen el contenido. Los autores recomiendan leer el capítulo completo, y usar las conclusiones como comprobación.

El orden de los capítulos sigue una lógica que ha demostrado su validez, al enfocar primero el aprendizaje en los aspectos teóricos para formar sobre esta base habilidades prácticas.

Cada capítulo del libro tiene un objetivo definido, que puede ir desde conocer sobre un tema hasta alcanzar habilidades prácticas, en este último caso los capítulos incluyen ejemplos que se resuelven completamente, explicándose las razones que provocan esas soluciones.

En el texto los conceptos formales se resaltan cambiando la inclinación de las letras y hay iconos para llamar la atención sobre determinados puntos.

Desde el punto de vista de una publicación electrónica, en la actualidad se desarrolla una nueva versión del libro electrónico ADOOSI (Análisis y Diseño Orientado a Objetos de Sistemas Informáticos), que responde a la metodología de igual nombre. El método DIBAO forma parte de esta nueva versión de la metodología, por lo que el diseño del nuevo producto de software incluye las especificaciones sobre el método con el nivel de detalle que se describen en el texto, pero le da una nueva dimensión a la información al auxiliarse de la tecnología de multimedia.

La idea es utilizar el libro electrónico como medio y método de enseñanza. Por si solo debe constituir un medio de enseñanza, mientras que por la forma en que será aplicado incidirá en el método de enseñanza que se emplee en su transmisión.

El libro electrónico pasará a formar parte de la bibliografía que poseen los estudiantes de la carrera de Ingeniería Informática en el país y está disponible por la red. Con ello se prevé lograr un cambio en la forma de estudio, propiciando un nuevo método de enseñanza. Su uso no sustituye en ningún momento el trabajo del profesor, pues éste seguirá transmitiendo sus conocimientos, pero el estudiante tendrá la posibilidad de estudiar de manera individual.

El libro podrá ser consultado para ampliar los conocimientos adquiridos en clases y, a través del módulo de evaluación, el estudiante comprobará si ha entendido los principios del desarrollo de un software utilizando los métodos descritos en el libro.

Experiencias de su utilización.

La propuesta de método de diseño de la BD que se describe en este trabajo se incluye dentro del curso de Diseño de Sistemas Informáticos, que reciben los estudiantes de 4to año de la carrera de Ingeniería en Informática, desde el curso 1999-2000. Además desde ese mismo período forma parte de los cursos de la metodología ADOOSI que se han impartido a empresas del país (AereoVaradero S. A - Transporte aéreo de carga y CEDISAP – desarrollo de software médico).

Para su impartición se combina la presentación de los contenidos a través de conferencias y consulta individual a los textos elaborados, con el desarrollo de talleres, de cuatro horas de duración, en los que se parte de una situación problemática, descrita en un caso de estudio, y se llega a la solución por iteraciones sucesivas en la que los estudiantes trabajan en grupos. La idea de trabajar en grupos pretende simular la situación real en la que en la solución de un determinado problema, por lo general está involucrada más de una persona y es importante tener en cuenta todos los criterios para llegar a una mejor solución.

La versión actual del método se ha aplicado a proyectos que han sido fundamentalmente sistemas de gestión implementados usando Borland Delphi para el trabajo con BDR (16) y de propósito general (7) para lo cual se ha utilizado la persistencia hacia fichero. En la actualidad el 49% de los proyectos, de estudiantes de 4to año de la carrera de Ingeniería Informática en el país, utilizan el método como parte de la metodología ADOOSI 5.0.

En el Departamento de Matemática Aplicada del Instituto Nacional de Ciencias Agrícolas se está utilizando el método para desarrollar un software que se incluye en una tesis de maestría próxima a defender.

Actualmente también se incluye este método de diseño dentro del curso de análisis y diseño que reciben los estudiantes de la carrera de Computación y Sistemas de la Universidad Peruana de Ciencias Aplicadas (Lima, Perú) y en la carrera de Sistemas Automatizados de Dirección del Instituto Técnico Militar José Martí (Cuba). En este último, este curso el libro fue utilizado como material de referencia en la impartición de la asignatura de Base de Datos. Su utilización permitió que por primera vez se impartiera, en la carrera de informática de este centro, temas relativos al diseño de la BD a partir de un MOO y profundizar en otros relacionados con los lenguajes de consulta para el modelo relacional.

En el caso del ISPJAE (Instituto Superior Politécnico José Antonio Echeverría) el texto recoge un poco más de los que se imparte en la docencia en cuanto al modelo relacional. Los estudiantes de 4to año de la carrera en Informática poseen el subconjunto correspondiente al capítulo 7 pues en ese año es en el que reciben estos contenidos. En ambos casos la estructura del libro ha servido de base para estructurar las asignaturas en las que se ha utilizado.

Gebase se encuentra en prueba en CEDISAP, en el desarrollo de aplicaciones reales.

Conclusiones.

El método DIBAO viene a suplir una de las necesidades de la industria de software actual en cuanto al diseño de la BD a partir del MOO ya que:

1. El modelo de persistencia propuesto tiene en cuenta todas las posibles alternativas, en cuanto a formas de almacenar la información, que se pueden encontrar en la etapa de desarrollo de un software, a diferencia de otras propuestas que profundizan en solo un aspecto o lo tratan de forma insuficiente.

2. Como parte del modelo de persistencia se han sistematizado los pasos a seguir en el diseño de la BD, así como recomendaciones que permiten obtener las especificaciones estáticas y dinámicas, utilizando los diagramas de clases y transición de estado y formatos para especificar algunas restricciones y fórmulas dinámicas que no se derivan de estos diagramas. Además se describe cómo obtener, de estos diagramas y especificaciones textuales, información útil sobre los objetos relacionada con su persistencia.
3. Se ha definido una capa persistente de clases para encapsular el trabajo con el medio de almacenamiento.
4. Se obtuvo un CASE que permite el diseño y la generación de la BD.

Los resultados de esta investigación ya han tenido aplicaciones concretas, en particular cuando se diseña para almacenar los datos en ficheros y en SQL Server, a partir del uso de Borland Delphi. Además se ha trabajado en su divulgación a través de cursos y confeccionado materiales.

Incluir toda esta información en un libro tanto en formato textual como digital, constituye un aporte importante en el proceso de capacitación de los futuros ingenieros y del personal que trabaja en las empresas.

Referencias bibliográficas

- [HER97] Hernández, A. “Soluciones al diseño de la base de datos a partir de un desarrollo orientado a objetos”. IX Conferencia Científica de Ingeniería y Arquitectura. II Taller de Informática Aplicada a la Educación Superior. Cuba. Diciembre 1997.
- [HER00a] Hernández, A. y Delgado, M. “Modelación de los comportamientos estáticos y dinámicos en el diseño de la base de datos a partir de un modelo orientado a objetos”. VII Convención Internacional Informática 2000, VII Congreso Internacional de Nuevas tecnologías y Aplicaciones Informáticas, Cuba. Mayo 2000.
- [HER00b] Hernández, A.; Mato, R. y Tenorio, D. “Base de Datos: coexistencia de los modelos relacional y orientado a objetos”. Instituto Técnico Militar José Martí. Septiembre 2000.
- [HER01] Hernández, A. y Alvarez, S. “Capa persistente para el almacenamiento de objetos”. Revista Ingeniería Industrial, XXII (1) Cuba. 2001.
- [PAS96] Pastor, O.; Pelechano, V.; Bonet, B. y Ramos, I. “OO-Method 2.0: una metodología de análisis y diseño orientado a objetos”. Reporte de investigación Departamento de computación y sistemas, Universidad Politécnica de Valencia. España. 1996.
- [PER01] Pérez, L. y Concepción, C. R. “Generación de la base de datos (Gibase versión 2.0)”. Trabajo para optar por el título de Ingeniería en Informática. Centro de estudios de Ingeniería y Sistemas, Instituto Superior Politécnico José Antonio Echeverría. Cuba. Junio 2001.